Combining Taxonomies using Word2vec

Tobias Swoboda, Matthias Hemmje
University of Hagen
Faculty for Mathematics and Computer Science
Universitätsstraße 47, 58085 Hagen, Germany
Tobias.Swoboda@fernuni-hagen.de,
Matthias.Hemmje@fernuni-hagen.de

Mihai Dascalu, Stefan Trausan-Matu
University Politehnica of Bucharest
Computer Science Department
313 Splaiul Independentei, Sector 6, Bucharest, Romania
mihai.dascalu@cs.pub.ro,
stefan.trausan@cs.pub.ro

ABSTRACT

Taxonomies have gained a broad usage in a variety of fields due to their extensibility, as well as their use for classification and knowledge organization. Of particular interest is the digital document management domain in which their hierarchical structure can be effectively employed in order to organize documents into content-specific categories. Common or standard taxonomies (e.g., the ACM Computing Classification System) contain concepts that are too general for conceptualizing specific knowledge domains. In this paper we introduce a novel automated approach that combines sub-trees from general taxonomies with specialized seed taxonomies by using specific Natural Language Processing techniques. We provide an extensible and generalizable model for combining taxonomies in the practical context of two very large European research projects. Because the manual combination of taxonomies by domain experts is a highly time consuming task, our model measures the semantic relatedness between concept labels in CBOW or skip-gram Word2vec vector spaces. A preliminary quantitative evaluation of the resulting taxonomies is performed after applying a greedy algorithm with incremental thresholds used for matching and combining topic labels.

Keywords

Word2Vec, taxonomy integration, ontology alignment, automated semantic integration

1. INTRODUCTION AND MOTIVATION

According to Berners-Lee et al. [1], ontologies are the foundation of the semantic web by conceptualizing different domains and by formally defining the relations among terms. In addition, *"the most typical kind of ontology for the Web has a taxonomy and a set of inference rules*." - Berners-Lee et al. [1]. Lexical taxonomies discriminate concepts (categories or classes), which can have multiple sub-classes (through the hypernym/hyponym relationships), further defining and refining these concepts. This generates a directed acyclic graph (DAG) as underlying representation of our taxonomy. The nodes of the DAG may be abstract and are machine-readable representations of the concept. The readability of concepts is increased by their linkage to labels that can be expressed in natural language. After building a coherent, taxonomical representation for a knowledge domain, the aim shifts towards retrieving relevant documents from an information system. Two major approaches emerge: querying and browsing [4]. Both benefit from the usage of taxonomies. Querying can be enhanced by *Named Entity Recognition* (NER), the process of finding text tokens that can be identified as named representations or labels of certain concepts within the taxonomy [7], p 761 ff. Providing content-based categories where each category is based on a concept from the taxonomy enables browsing. This provides a hierarchy of categories and sub-categories to browse for content.

Now the research question arises: How can we effectively build new taxonomies to facilitate automated document classification and retrieval by combining existing taxonomies with specific seed concepts? To start with, a rather prominent taxonomy, commonly used for organizing scientific papers in the knowledge domain of computer science, is the 2012 ACM Computing Classification System (CCS) [6]. Taxonomies organizing large text corpora like the entire set of ACM publications are faced with the problem of creating meaningful relations between the underlying concept hierarchies and of supporting automated text categorization (TC) [12]. In the remainder of this paper, the terms concept and category are used synonymously.

Our aim consists of reusing knowledge from widely adopted standard taxonomies, combining them and facilitating automated document categorization while managing novel or specific knowledge domains or document collections. Our research is conducted in the context of two very large European H2020 research projects – RAGE [15] and EDISON [8] –, but our method and the obtained results are directly applicable to any knowledge domain. Both projects are using digital libraries providing access to knowledge resources, generated by and relevant for these projects. Taxonomies are used to organize and automatically categorize the documents specific to the scope of each project. We were faced with a major drawback of existing and established taxonomies that were either too broad or too extensive, while providing little insights in terms of effectively classifying documents. Large and general taxonomies would have been inappropriate to use within the projects' digital libraries because the majority of collections would have been either over-populated or empty, thus defeating the purpose of equitable content-based categories as presented in Figure 1.

This paper outlines our approach to reuse parts of well-accepted standard taxonomies in order to create domain specific taxonomies. As a specific example, we have opted to focus on the RAGE taxonomy because a small and specific seed taxonomy has already been specifically created for the RAGE project.

DocEng '16, September 12-16, 2016, Vienna, Austria

DOI: http://dx.doi.org/10.1145/2960811.2967151

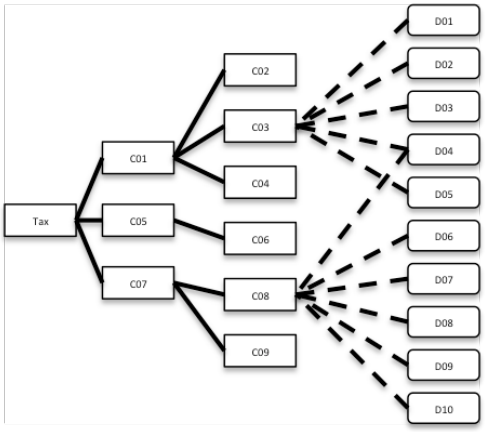

Figure 1. Illustration of an unsuitable taxonomy

In terms of structure, section two presents an evaluation of existing approaches, followed by the presentation of our model and of our generalizable combination algorithm. In contrast to other existing approaches, our method relies on measuring the semantic relatedness between vector representations of taxonomy labels in continuous bag of words (CBOW) or skip-gram Word2vec models [10]. Afterwards, preliminary results are presented, followed by discussion and conclusions.

2. STATE OF THE ART

2.1 Approaches to combine taxonomies

When generating taxonomies from scratch, there are two fundamental strategies: a) manual taxonomy construction and b) automated machine learning and the use of specific natural language processing tools. The manual approach is a cumbersome labor-driven process performed by domain experts or taxonomy engineers. On the upside, the generated taxonomy is subjectively representative for the people who generated it. The United States National Information Standards Organization (NISO) suggests two fundamental strategies when manually generating taxonomies [11]. *Top-down*: A committee of experts selects the broadest terms of a knowledge domain and connects narrower terms with these until a desired level of specificity is reached. *Bottom-up*: A committee of experts starts with a set of terms related to the knowledge domain and aggregates them from narrow terms to more general terms. In a nutshell, although the resulted taxonomy represents a coherent shared view of the domain, the manual generation of taxonomies is a highly time-consuming undertaking. One requires groups of experts to collaborate and agree on a common representation of knowledge.

The alternative approach of automated taxonomy construction requires sample texts and a set of keywords for machine learning algorithms to learn from. For example, Gollub et al. [5] propose the dynamic taxonomy generation based on search terms used during querying. Their approach dynamically updates the utilized taxonomy based on search terms and the amount of documents associated with a given concept.

The main challenge arises: provide suitable texts and adequate keywords. Depending on the knowledge domain and algorithm, the lack of available documents and keywords can lead to results with a limited beneficial impact [9]. However, none of the available automated techniques is applicable for our needs because neither the document set in question, nor a set of search terms are previously available. The remaining challenge lies in combining the small seed taxonomies with sub-trees of the commonly accepted big taxonomy, without having a set of documents or search terms as reference.

From a broader perspective, the combination of two taxonomies and, in general, ontologies, is an extremely difficult process called ontology matching or alignment [13]. This process involves the modification of the content and structure of both ontologies. Our approach considers a simpler case, in which only one ontology is modified and the changes represent only additions of concepts. Moreover, we work only with the taxonomic backbone of ontologies. However, based on the proposed method and our findings presented in detail in the following sections, we consider that our approach may be extended for ontology mapping.

2.2 Word2vec

Multiple semantic models used to evaluate the relatedness between concepts and/or documents have been proposed in time, ranging from traditional vector spaces (e.g., Latent Semantic Analysis, LSA) [10], probabilistic models (most notable, Latent Dirichlet Allocation, LDA) [2], or the newly introduced Word2vec model based on neural networks [10]. Although all semantic models enable the assessment of similarity between concepts, we opted to rely on Word2vec, as its reported accuracy on the sentence completion task in the Microsoft challenge was highest of all models (58.9% accuracy for determining the correct/appropriate word to be introduced within a sentence) [10]. This emphasizes the fact that Word2vec is one of the most suitable automated models for building coherent representations and for creating context-driven word associations, central elements within our task of combining taxonomies in a coherent overall representation suitable for the domain.

Word2vec uses neural networks to generate high dimensional vector representations for each word or document. Neural networks usually require labeled input-output pairs to learn, but these associations cannot be provided by a flat text. In order to address this limitation, two alternatives have been introduced [10]. First, *CBOW* (continuous bag of words) predicts a word given its context. Therefore, the words before and after every instance of a word are used as input in a training sample expecting this word as output. The second alternative, *skip-gram* works the other way around: it takes single words as input samples, while the surrounding words are the expected output. Cosine similarity can then be used to assess the degree of similarity between words.

Interesting linguistic properties in the arithmetic manipulation of the resulting vectors have been previously shown [10]. Relationships between word vectors are encoded by their offset in the generated high dimensional space. This way, for example, gender is a certain offset that can be applied to the vector representation of "*boy*" in order to get a vector very close to the vector representation of "*girl*". We used these geometric regularities induced by high cosine similarities to compute the relevance of concepts in other taxonomies.

3. METHOD

3.1 Corpus

The 2012 ACM Computing Classification System's (CCS) [6] wide acceptance is largely a result of its critical review and subsequent revision process. In the RAGE project, an initial small seed-taxonomy has already been developed. This taxonomy is highly specific in its field, but not widely accepted outside the RAGE project. The subsequently described model explains our approach to combine parts of widely accepted taxonomies with our own highly specific seed taxonomy.

3.2 Model

Our model is based on the representation of taxonomies $T = \{C, E, L\}$ as directed acyclic graphs (DAGs). These DAGs consist of nodes - concepts (C) and directed hypernym/hyponym relationships $E \subset C \times C$ between the concepts. L is the set of labels for the given concepts.

As a starting point, we consider two given taxonomies: a general one $TG = \{CG, EG, LG\}$, and a specific or seed taxonomy: $TS = \{CS, ES, LS\}$. The resulting new taxonomy is denoted as $TN = \{CN, EN, LN\}$. Because everything in the seed taxonomy is deemed relevant, the resulting new taxonomy is initialized as $CN = CS$, $EN = ES$ and $LN = LS$. Two possible cases for the general concepts $c \in CG$ and e (the connecting edge between both DAGs) have been identified:

- *Case 1*: c is an inner node of TG that is semantically relevant, yet still unutilized in TS. In this case c and all its descendants $CU \subset CG$ along with their Labels $LU \subset LG$ and inner edges $EU \subset EG$ are integrated into TS resulting in $TN = \{CN \cup CU, EN \cup EU \cup \{e\}, LN \cup LU\}$.
- *Case 2*: c is a leaf of TG that is semantically relevant, yet still unutilized in TS. In this case, c along with its labels $LU \subset LG$ are integrated into TN resulting in $TN = \{CN \cup \{c\}, EN \cup \{e\}, LS \cup LU\}$.

In both cases, the edge e integrates the suitable subgraph with TS at its most appropriate place. This approach is illustrated in Figure 2 in which CG7, an inner node, is linked with the root of TS_v while CG2, a leaf, is linked with CS4.

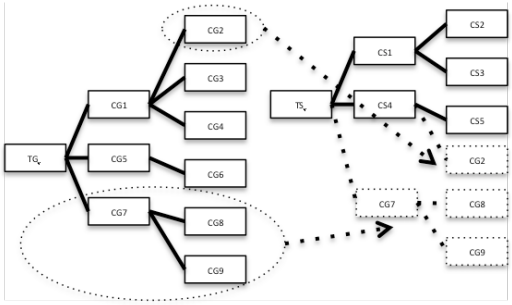

Figure 2. Taxonomy combination

In order to perform the taxonomy integration, the system must determine for every given concept $c \in CG$ if each concept and all its sub-concepts are relevant for the knowledge domain of TS. In case the concept is relevant, the system must also determine which is the best concept from CS to link it to. This evaluation can be alternatively performed manually, but this would have been a cumbersome task as the ACM 2012 CCS has 2299 concepts.

3.3 DFS-based Algorithm

Our approach is essentially a modified depth first search (DFS) through TG [3]. Whenever a concept $c \in CG$ is determined to be relevant for TS, it and all its descending concepts are linked to the most relevant concept $c_s \in CS$. The relevance values of the descendant nodes of c are not computed because they have already been added to TS. We used Word2vec to compute whether a concept $c \in CG$ is relevant for TS and what are the concepts $c_s \in CS$ to which it is linked. Therefore, vector representations for every concept in all taxonomies were generated. Stop words (i.e., natural language words with limited meaning, such as "the", "and", "a", etc.) and punctuation characters of the labels were removed. Afterwards, Word2vec vector representations of every word in the labels for one category were aggregated in order to compute the concept representation in the high dimensional space as the geometric mean of all word vectors. A concept $c \in CG$ was deemed relevant for TS if its cosine similarity to any $c_s \in CS$ is between configurable upper and lower thresholds. This essentially generates relevance radii around c. A concept $c_s \in CS$ with a cosine similarity value outside these margins would not render c relevant because c is either too similar or too different from these concepts.

After generating a list of concepts $c_s \in CS$ within the threshold, our algorithm uses a Greedy approach [3] to attach c to the concept c_s that has the highest cosine similarity. We additionally implemented a limitation of how many descendant concepts c could have to be relevant. This limited the size of the reusable parts of TG. Without such a restriction in place, the system could for example attach the root of TG to TN and finish after one step. The resulting taxonomy would be too general for the knowledge domain of TN and essentially defeat the purpose of our approach. The next section describes the implementation of this method.

3.4 Word2vec based implementation

The RAGE seed taxonomy models the knowledge domain of applied gaming, it is denoted CS in accordance to the algorithm and contains 46 concepts, out of which the top level categories reflect: *assessment*, *decision-making and socio-emotional behavior*, *embodiment and physical interaction*, *emotion detection*, *evaluation*, *game balancing and personalized learning*, *interaction data and exchange and storage*, *interactive storytelling*, *natural language* and *social gamification*.

Before running our algorithm for combining the RAGE seed taxonomy with the ACM taxonomy, we generated vector representations of vocabularies that were then loaded in our software. Word2vec was used with both CBOW and the continuous skip-gram approaches to generate 200 dimensional word vectors. The following training sets were used: A dump of Google news articles, retrieved January 18, 2016 from http://mattmahoney.net/dc/text8.zip and the first billion characters of Wikipedia dump, retrieved January 23, 2016 from http://mattmahoney.net/dc/enwik9.zip. Our approach compares the similarities between category labels; therefore, it does not require sample documents that were already assigned to specific categories. Depending on the training set, our system was able to generate vector representations for a different number of categories. When using word vectors learned from Google news, our system was able to generate vector representations for 2230 concepts from 2299 concepts of CG. After using the first billion characters of Wikipedia as training set, the system was able to generate vector representations for 2266 from the 2299 concepts. The difference is due to the fact that the remaining ACM concepts have labels that have no vector representations, as the underlying words from the labels were not part of the training set vocabulary. Overall, word vectors based on the skip-gram approach yielded higher cosine similarity values. The same is true for the Wikipedia training set over the Google news training set. There were some particular cases. For example, the ACM taxonomy contains 50 concepts with labels containing the term "*analysis*" that all had a high cosine similarity with the RAGE concept "*Assessment dashboard and analysis*".

4. PRELIMINARY RESULTS

We ran multiple experiments with different configurations. As previously described, we used four different vector representations for the vocabulary of the English language. With each of these, the lower cosine similarity threshold was increased

in 0.05 intervals. The upper threshold was set to the maximum possible value: 1. As the seed taxonomy had 46 concepts, a threshold of 20 maximum descendants seemed a reasonable size for the sub-trees to be transferred to the new taxonomy.

Multiple properties for the new taxonomy TN were measured consisting of: the number of concepts, of leafs and of connections in the taxonomy. For the latter, the amount of connections indicates for how many concepts $c_g \in CG$ of the ACM taxonomy, concepts within the thresholds could be identified within the seed taxonomy $c_s \in CS$. All experiments show, that the amount of connections decreases with an increasing lower threshold while maintaining a high amount of concepts and leafs until a certain point. This is due to the usage of entire sub-taxonomies when a common inner concept is deemed relevant. For threshold values higher than this point, which differs based on algorithm and training set, concepts and leafs begin to decrease. Obviously, the more connections are found, the more concepts from the general taxonomy have a chance to be relevant within the newly generated taxonomy. Depending on the used training set and algorithm, the most adequate taxonomies were generated by imposing a minimum threshold of .6 to .7. These taxonomies contained concepts like *acceptance testing*, *interactive simulation*, *graphics input devices* and *network games*, while remaining small without essentially transferring most of the concepts from the ACM taxonomy to the RAGE seed taxonomy. However, some distant concepts for serious/applied gaming were considered (e.g., *distributed memory*) due to the multiple meanings and senses that concepts like "memory" can have (working memory linked to learner comprehension versus computer memory).

5. DISCUSSION AND CONCLUSIONS

Our approach extends a seed taxonomy by selecting the most semantically related concepts of a general taxonomy and adding them into the seed taxonomy. Because only the seed taxonomy is modified by the addition of new concepts, our task is simpler than that of ontology alignment [13]. However, the underlying concept of projecting concepts as vectors into high dimensional spaces in order to derive their similarities can alternatively be used for a variety of applications like the automated alignment of ontologies and semantic integration. We must also present some limitations induced by the fact that all information about concepts is derived from their labels. By relying only on these few words in order to map a concept into the high dimensional space, we were faced with problems in terms of synonyms and homonyms. Hypernyms and hyponyms are automatically addressed by considering the hierarchical structure of the taxonomy during its generation. In addition, we must highlight another intrinsic limitation as many ACM concepts contain the terms *analysis*, *assessment* or *evaluation* in their labels. Most of them were matched to the RAGE concept *assessment* or *assessment dashboard and analysis*. Overall, the similarities between vocabulary labels induced a higher degree of relatedness and many of the concept associations made sense while relating to human expertise. However, there were some associations that need to be manually cleaned.

Our approach is, to the best of our knowledge, unique as it only relies on the concept labels without requiring query terms, sample documents or domain expert information. A disadvantage of our approach lies in the fact that the available information is limited to the labels of the available concepts. This means that potentially inadequate concepts, with similar labels, can be selected. Therefore, these automatically generated taxonomies are best used to speed up the manual taxonomy generation, by providing potential candidates to domain experts.

In future works, the document corpora for the RAGE and EDISON projects will be curated and automated text categorization will be applied on all documents. Expert interviews will be conducted to evaluate and manually refine the generated taxonomies and provide in-depth validations and an effectiveness assessment. In terms of comparisons, alternative semantic word-vector models will be employed within the proposed approach.

6. ACKNOWLEDGMENTS

This work was partially funded by the EC H2020 RAGE (Realising and Applied Gaming Eco-System) No. 644187 and the EC H2020 EDISON No. 675419 projects.

7. REFERENCES

[1] Berners-Lee, T., Hendler, J., Lassila, O. 2001. The semantic web. *Scientific American Magazine*: pp. 35-44, May 2001

[2] Blei, D. M., Ng, A. Y., Jordan, M. I. Latent Dirichlet Allocation, *Journal of Machine Learning Research 3, pp. 993-1022, 2003*

[3] Cormen, Th. C., Leiserson, C. E., Rivest, R., Stein, C. 2001. *Introduction to Algorithms*. Second Edition, MIT Press, Massachusetts, USA, 2001

[4] Cox, K. 1992. Information Retrieval by Browsing. *Proceedings of The 5th International Conference on New Information Technology*, Hong Kong, 1992

[5] Gollub, T., Volkse, M., Hagen, M., Stein, B. 2014. Dynamic taxonomy composition via keyqueries. *IEEE/ACM Joint Conference on Digital Libraries (JCDL)*, pp. 39-48, London

[6] The 2012 ACM Computing Classification System, Retrieves December 28, 2015 from Association for Computing Machinery, Inc., New York, NY

[7] Jurafsky, D., Martin, J. H. 2009. *Speech and language processing. An introduction to natural language processing, computational linguistics and speech recognition*. 2nd edition, Upper Saddle River, N.J., London: Pearson Prentice Hall

[8] Konijn, J. 2015. Education for Data Intensive Science to Open New science frontiers (EDISON) – Project proposal

[9] Liu, X., Song, Y., Liu, S., Wang, H. 2012. Automatic Taxonomy Construction from Keywords, *ACM SIGKDD conference*, August 12-16, Beijing, China

[10] Mikolov, T., Chen, K., Corrado, G., Dean, J. 2013. Efficient Estimation of Word Representation in Vector Space. *Proceedings of Workshop at ICLR*. Retrieved December 29, 2015 from http://arxiv.org/pdf/1301.3781.pdf

[11] National Information Standards Organization (NISO) 2005. *Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies*.

[12] Sebastiani, F. 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys* vol. 34, pp. 1-47

[13] Shvaiko, P., Euzenat, J. 2013. Ontology Matching: State of the Art and Future Challenges. *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp.158-176

[14] Stein, B., Gollub, T., Hoppe, D. 2011, Beyond Precision @ 10: Clustering the Long Tail of Web Search Results, *20th ACM International Conference on in Information and Knowledge Management*, Glasgow, UK, pp. 2141-2144

[15] Westera, W. 2014, *Realising an Applied Gaming Ecosystem* (RAGE) - Annex 1 to the Grant Agreement (Description of the Action) Part B